Feature Extraction

Dr. Amira Ashour

March 20, 2018

Content

- Introduction
- Feature Extraction
  - Shape Features - Binary Object Features
  - Histogram Features
  - Color Features
  - Spectral Features
- Feature Analysis
  - Feature Vectors and Feature Spaces
  - Distance and Similarity Measures

Introduction

- The goal in image analysis is to extract useful information for solving application-based problems.
- The first step to this is to reduce the amount of image data using methods, such as: segmentation and filtering in the frequency domain.
- The next step would be to extract features that are useful in solving computer imaging problems.
- What features to be extracted are application dependent.
- After the features have been extracted, then analysis can be done.

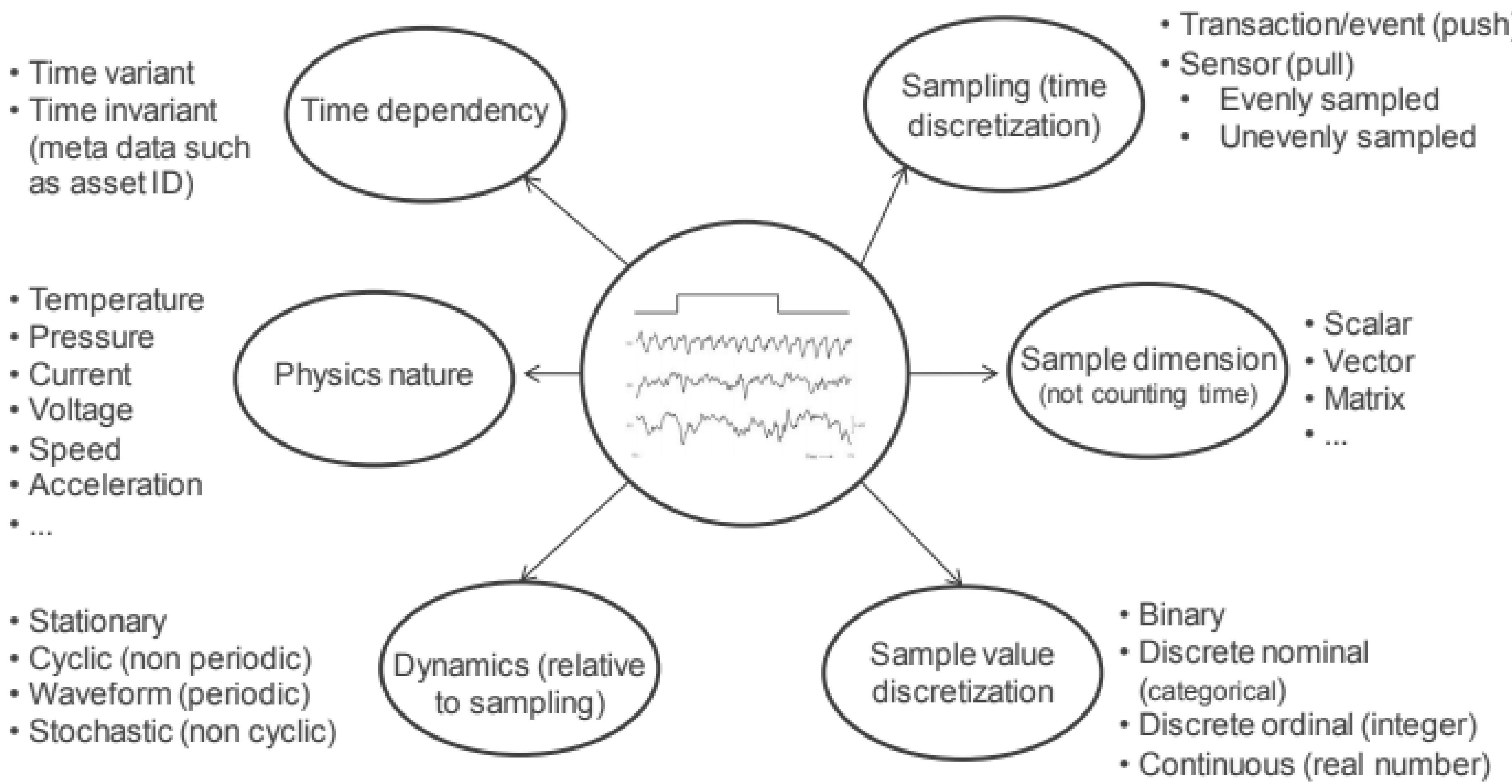
Feature extraction

- Feature extraction involves finding features of the segmented image.

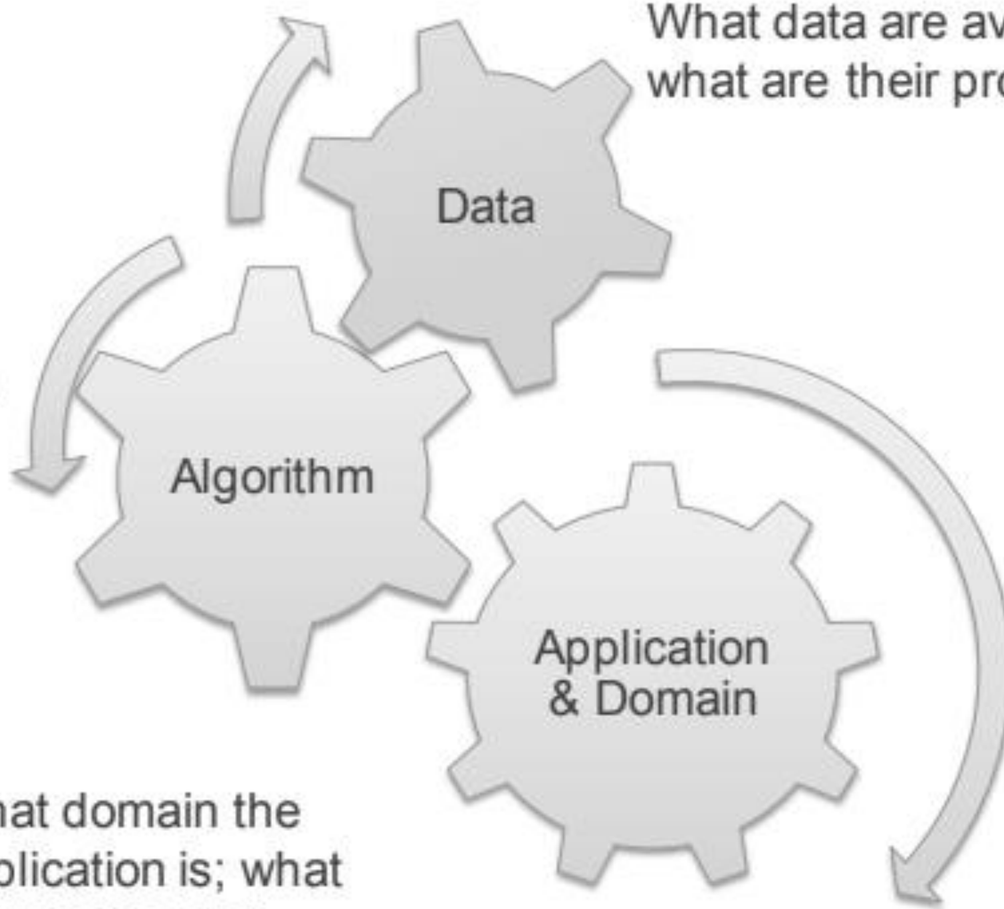
- Usually performed on a binary image produced from a thresholding operation.

- Common features include:
  1. Area.
  2. Perimeter.
  3. Center of mass.
  4. Compactness.

Data (signal) properties



What features to extract? Factors to consider...



Feature Extraction - method of capturing visual content of images for indexing & retrieval.

-***visual features***
(primitive or low-level image features)

Domain-specific features:
- fingerprints, human faces

General features:
- color, texture, shape

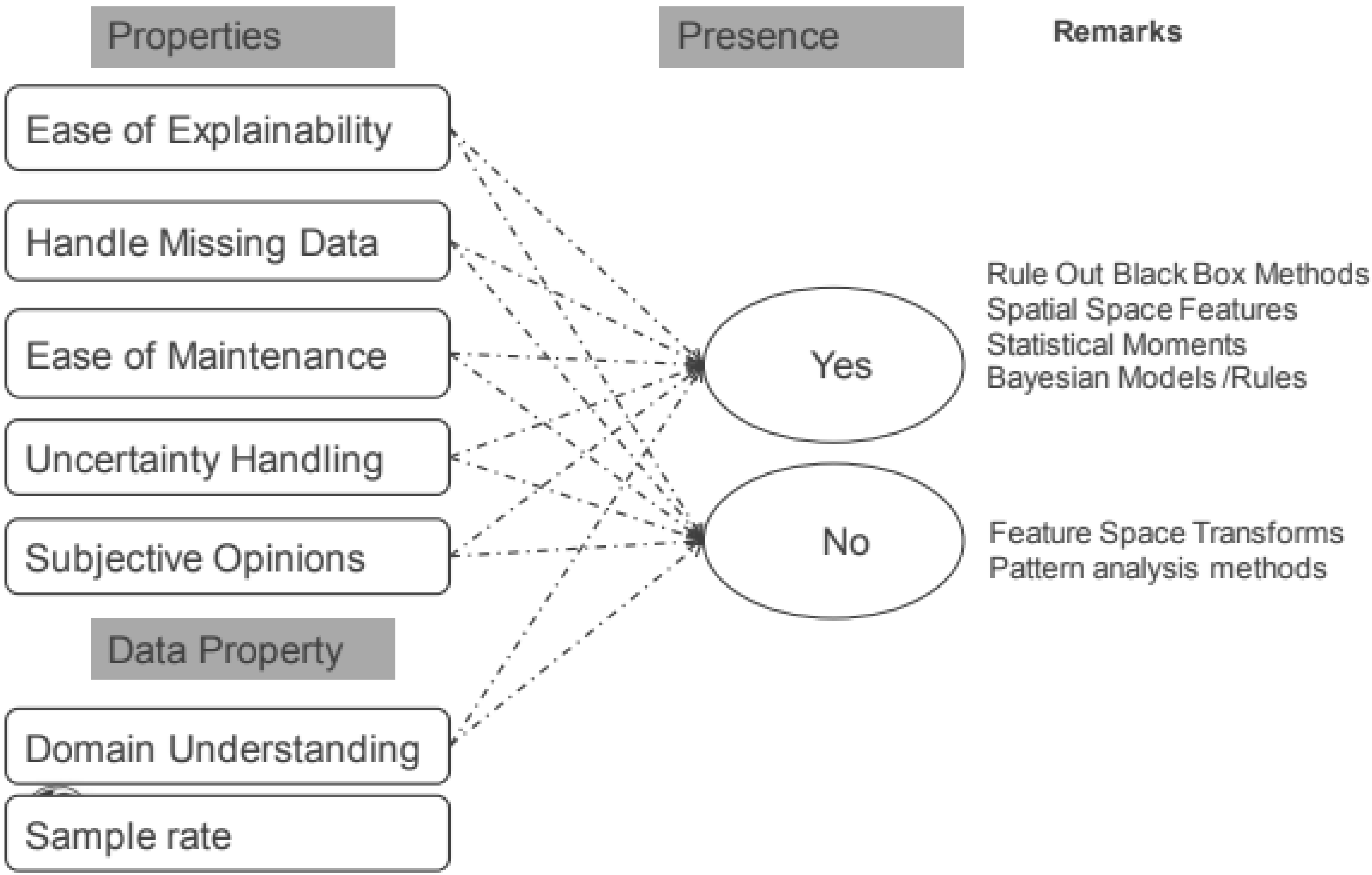
The issue of choosing the features to be extracted should be guided by the following concerns:

- the features should carry enough information about the image and should not require any domain-specific knowledge for their extraction.

- they should be easy to compute in order for the approach to be feasible for a large image collection and rapid retrieval.

- they should relate well with the human perceptual characteristics since users will finally determine the suitability of the retrieved images.

Properties of extracted features



Feature extraction method overview

- Data descriptive statistics
  - For sensors: RMS, variance, kurtosis, crest factor, correlation coefficient, ...
  - For events: count, occurrence rate, duration, time delays, ...
- Data descriptive models
  - Distribution models: Parametric distributions, histogram, ...
  - Information-based models: mutual information, minimal description length, ...
  - Regression models (use model parameters or modeling errors): curve fitting, AR models, ...
  - Classification/clustering models (use class label as feature), sequence matching likelihood
- Time-independent transforms
  - Explicit mathematical operations: difference, summation, ratio, logarithm, power n, ..
  - Principal component analysis, Independent component analysis, etc.
- Time series transforms (mainly for waveform signal)
  - Frequency domain, time-frequency domain, wavelet domain, EMD
- Domain dependent feature extraction
  - Physics based features: expected input-output or output-output relations, derived hidden states, etc.
  - Special procedures for data processing: operational regime segmentations, envelop analysis, etc.

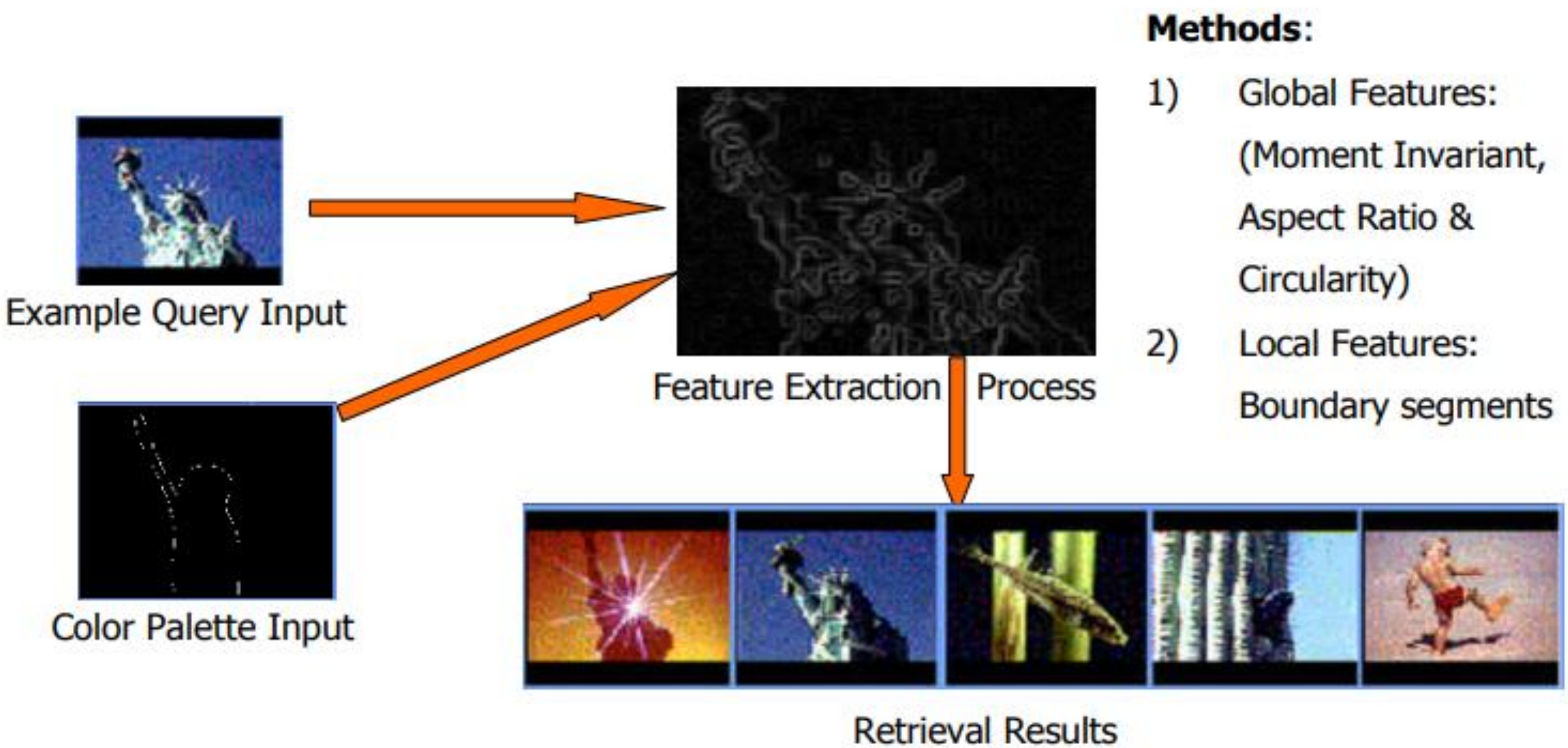
Texture Features

- Textures can be rough or smooth, vertical or horizontal etc
- Generally they capture patterns in the image data (or lack of them), e.g. *repetitiveness* and *granularity*

- Texture features:
  - Statistical measures:
    - Entropy
    - Homogeneity
    - Contrast
  - Wavelets
  - Fractals

Shape Features

- Depend on a silhouette (outline) of an image
- All that is needed is a binary image



Binary Object Features

- In order to extract the object features, we need an image that has undergone image segmentation and any necessary morphological filtering.

- This will provide a clearly defined object which can be labeled and processed independently.

- After all the binary objects in the image are labeled, each object can be treated as a binary image.
  - The labeled object has a value of ‘1’ and everything else is ‘0’.

Binary Object Features: Geometrical features of binary objects

- The labeling process goes as follows:
  - Define the desired connectivity.
  - Scan the image and label connected objects with the same symbol.
- After labeling the objects, an image filled with object numbers is produced.
- This image is used to extract the features of interest.
- Among the binary object features include area, center of area, axis of least second moment, perimeter, Euler number, projections, thinness ration and aspect ratio.

Binary Object Features

- In order to extract those features for individual object, we need to create separate binary image for each of them.

- This is done by assigning 1 to pixels with the specified label and 0 elsewhere.
  - If after the labeling process we end up with 3 different labels, then we need to create 3 separate binary images for each object.

Binary Object Features – Area

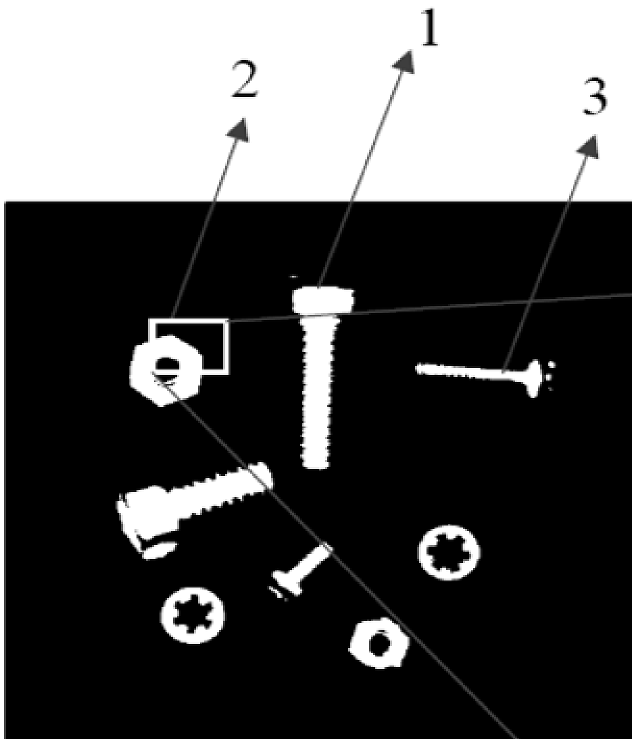
- The area of the *i*th object is defined as follows:

$$A_i = \sum_{r=0}^{height-1} \sum_{c=0}^{width-1} I_i(r,c)$$

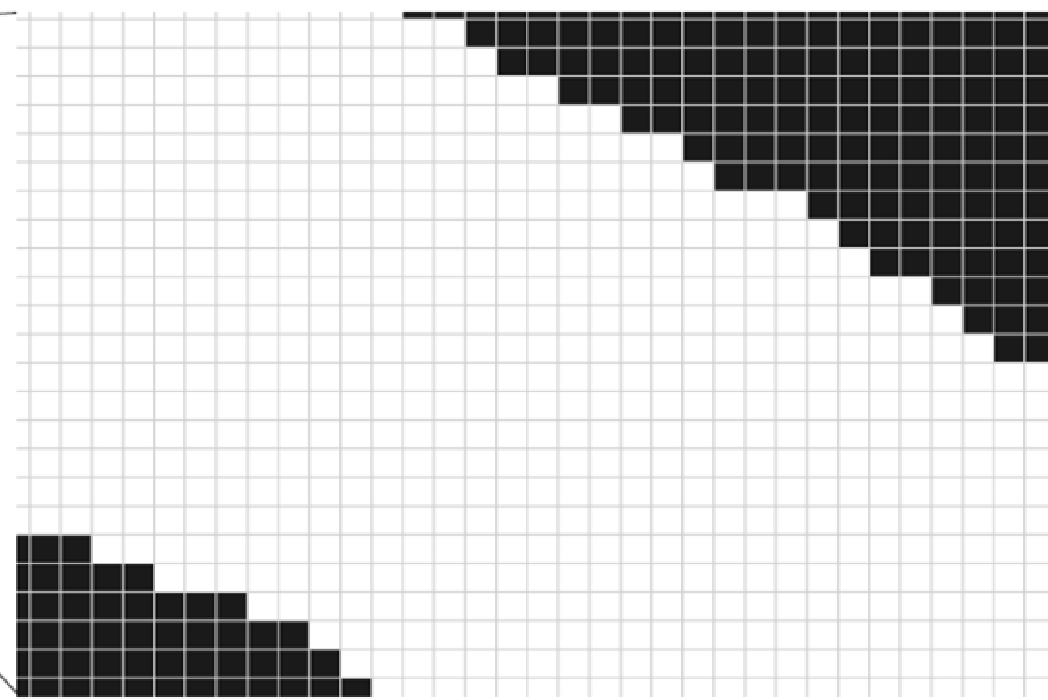
- The area A_i is measured in pixels and indicates the relative size of the object.

Feature extraction - area

Count the number of pixels in each labeled blob.



Simple algorithm – scan images and count the number of pixels labeled with a “2”



```
for(i=0;i<width;i++){
   for(j=0;jheight;j++){
      if(image[i][j]==2
         area[2] += 1;
   }
}
```

Binary Object Features – Center of Area

- The center of area is defined as follows:

$$\bar{r}_i = \frac{1}{A_i} \sum_{r=0}^{height-1} \sum_{c=0}^{width-1} r I_i(r,c)$$

$$\bar{c}_i = \frac{1}{A_i} \sum_{r=0}^{height-1} \sum_{c=0}^{width-1} c I_i(r,c)$$

- These correspond to the row and column coordinate of the center of the *i*th object.

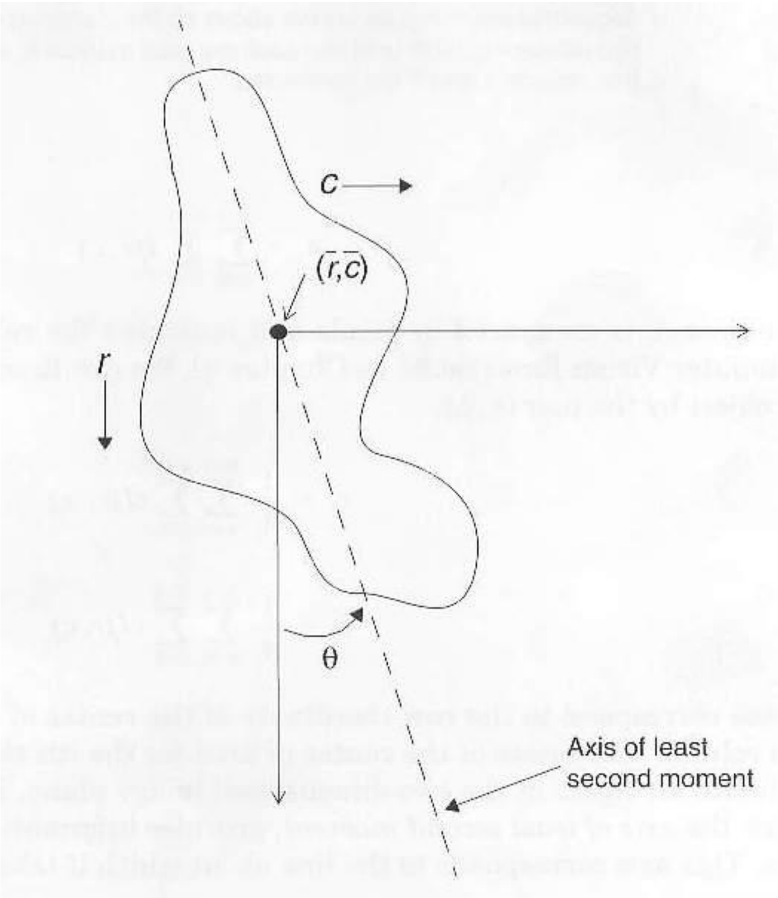
Binary Object Features – Axis of Least Second Moment

- The Axis of Least Second Moment is expressed as θ - the angle of the axis relatives to the vertical axis.

$$\theta_i = \frac{1}{2}\tan^{-1}\left(\frac{2\sum_{r=0}^{height-1}\sum_{c=0}^{width-1}(r-\bar{r})(c-\bar{c})I_i(r,c)}{\sum_{r=0}^{height-1}\sum_{c=0}^{width-1}(r-\bar{r})^2 I_i(r,c) - \sum_{r=0}^{height-1}\sum_{c=0}^{width-1}(c-\bar{c})^2 I_i(r,c)} \right)$$

Binary Object Features – Axis of Least Second Moment

- This assumes that the origin is as the center of area.
- This feature provides information about the object's orientation.
- This axis corresponds to the line about which it takes the least amount of energy to spin an object.



Binary Object Features - Perimeter

- The perimeter is defined as the total pixels that constitutes the edge of the object.

- Perimeter can help us to locate the object in space and provide information about the shape of the object.

- Perimeters can be found by counting the number of ‘1’ pixels that have ‘0’ pixels as neighbors.

Binary Object Features - Perimeter

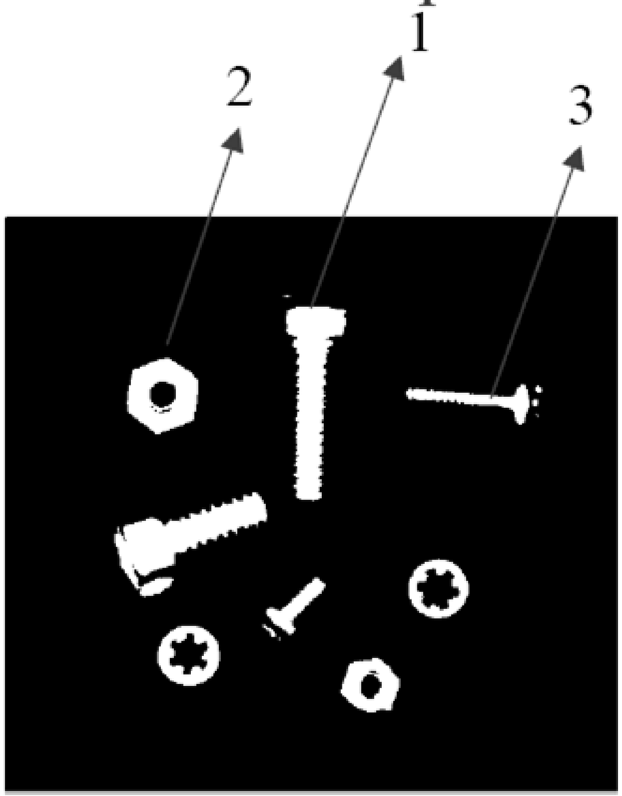
- Perimeter can also be found by applying an edge detector to the object, followed by counting the ‘1’ pixels.

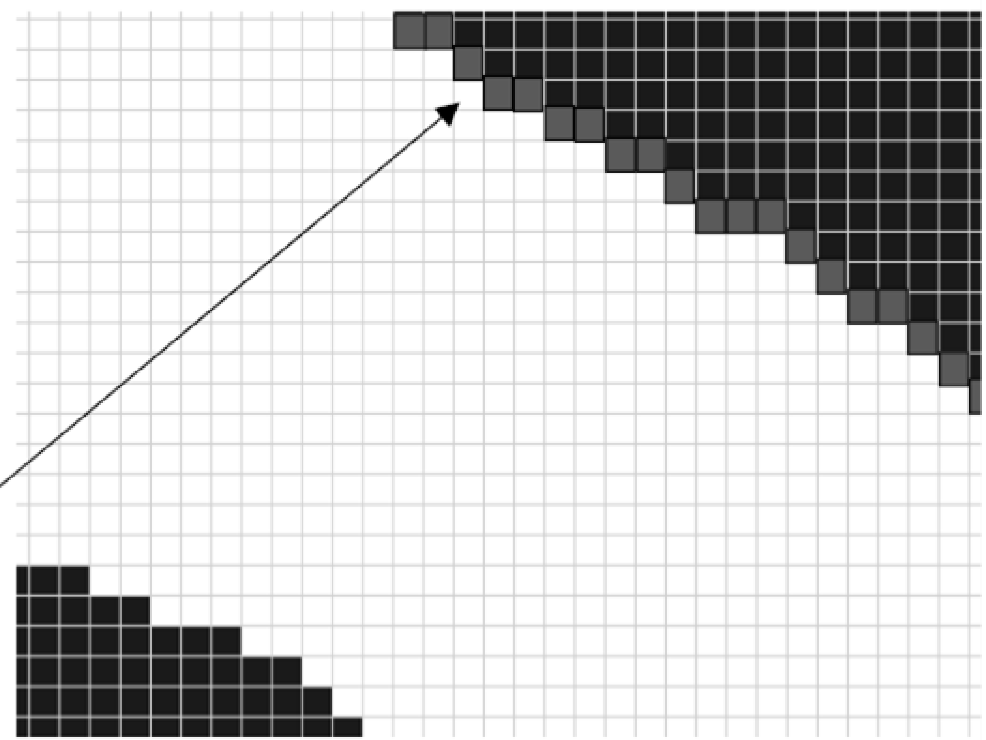
- The two methods above only give an estimate of the actual perimeter.

- An improved estimate can be found by multiplying the results from either of the two methods by $\pi/4$.

Feature extraction - perimeter

The number of pixels in a blob that border a “0”





Perimeter pixels

Feature extraction - compactness

- Other features can be obtained from simple measurements.

- Compactness = permiter^2/4πarea
- For a perfect circle
  - Compactness $= (2\pi r)^2 / 4\ \pi(\pi r^2)$
    $= 4\pi^2 r^2 / 4\pi^2 r^2 = 1$

- Long skinny blobs will have a high value for compactness.

Binary Object Features – Thinness Ratio

- The thinness ratio, *T*, can be calculated from perimeter and area.
- The equation for thinness ratio is defined as follows:

$$T_i = 4\pi\left(\frac{A_i}{{P_i}^2}\right)$$

Binary Object Features – Thinness Ratio

- The thinness ratio is used as a measure of roundness.
  - It has a maximum value of 1, which corresponds to a circle.
  - As the object becomes thinner and thinner, the perimeter becomes larger relative to the area and the ratio decreases.

Binary Object Features – Irregularity Ratio

- The inverse of thinness ration is called compactness or irregularity ratio, $1/T$.
- This metric is used to determine the regularity of an object:
  - Regular objects have less vertices (branches) and hence, less perimeter compare to irregular object of the same area.

Binary Object Features – Aspect Ratio

- The aspect ratio (also called elongation or eccentricity) is defined by the ratio of the bounding box of an object.

- This can be found by scanning the image and finding the minimum and maximum values on the row and column where the object lies.

Binary Object Features – Aspect Ratio

- The equation for aspect ratio is as follows:

$$\frac{c_{\max} - c_{\min} + 1}{r_{\max} - r_{\min} + 1}$$

- It reveals how the object spread in both vertical and horizontal direction.

- High aspect ratio indicates the object spread more towards horizontal direction.

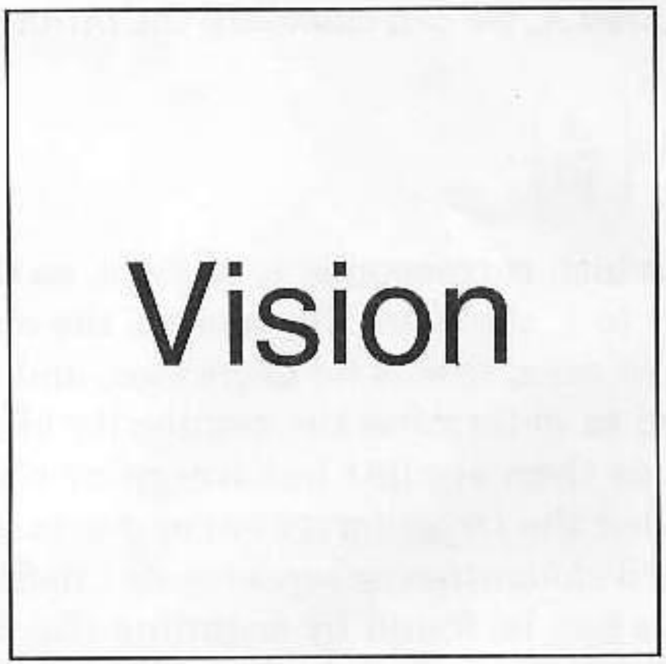
Binary Object Features – Euler Number

- Euler number is defined as the difference between the number of objects and the number of holes.
  - Euler number = num of object – number of holes

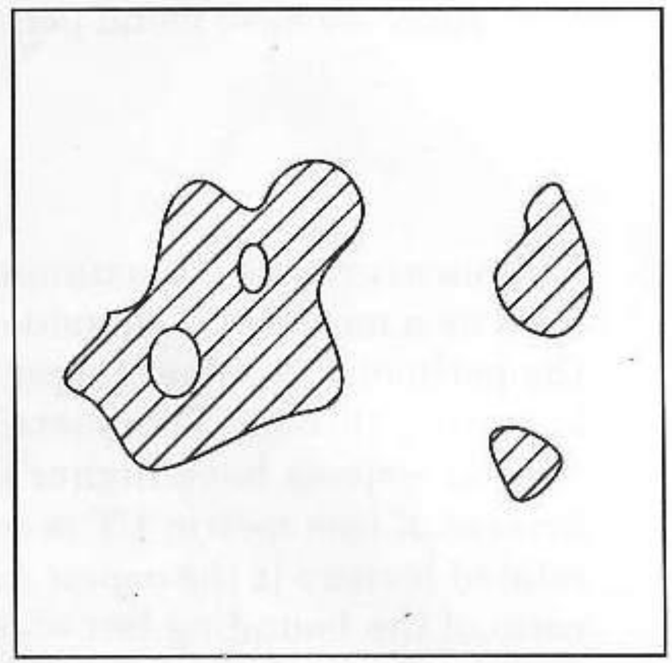
- In the case of a single object, the Euler number indicates how many closed curves (holes) the object contains.

Binary Object Features – Euler Number

- Euler number can be used in tasks such as optical character recognition (OCR).



a. This image has eight objects and one hole, so its Euler number is $8 - 1 = 7$. The letter V has Euler number of 1, $i = 2$, $s = 1$, $o = 0$, and $n = 1$.



b. This image has three objects and two holes, so the Euler number is $3 - 2 = 1$.

Binary Object Features – Euler Number

- Euler number can also be found using the number of convexities and concavities.
  - Euler number = number of convexities – number of concavities

- This can be found by scanning the image for the following patterns:

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Convexities

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

Concavities

Binary Object Features – Projection

- The projection of a binary object, may provide useful information related to object's shape.

- It can be found by summing all the pixels along the rows or columns.
  - Summing the rows give horizontal projection.
  - Summing the columns give the vertical projection.

Binary Object Features – Projection

- We can defined the horizontal projection $h_i(r)$ and vertical projection $v_i(c)$ as:

$$h_i(r) = \sum_{c=0}^{width-1} I_i(r,c)$$

$$v_i(c) = \sum_{r=0}^{height-1} I_i(r,c)$$

- An example of projections is shown in the next slide:

Binary Object Features – Projection

									$h(r)$
0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	1	0	0	0	2
0	0	1	1	1	1	1	0	0	5
0	1	1	1	1	1	1	0	0	6
0	0	0	0	0	1	1	0	0	2
0	0	0	0	0	0	0	0	0	0
0	1	2	2	4	4	3	0	0	

$c \longrightarrow$ (columns), $r \downarrow$ (rows), $v(c) \longrightarrow$ (last row), $h(r)$ (last column)

To find the projections, we sum the number of 1s in the rows and columns.

Other Features

Table 1. Representative shape descriptors

$$\text{Formfactor} = \frac{4\pi \cdot \text{Area}}{\text{Perimeter}^2}$$

$$\text{Roundness} = \frac{4 \cdot \text{Area}}{\pi \cdot \text{Max Diameter}^2}$$

$$\text{Aspect Ratio} = \frac{\text{Max Diameter}}{\text{Min Diameter}}$$

$$\text{Elongation} = \frac{\text{Fiber Length}}{\text{Fiber Width}}$$

$$\text{Curl} = \frac{\text{Length}}{\text{Fiber Length}}$$

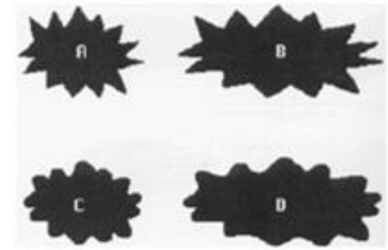
$$\text{Convexity} = \frac{\text{Convex Perimeter}}{\text{Perimeter}}$$

$$\text{Solidity} = \frac{\text{Area}}{\text{Convex Area}}$$

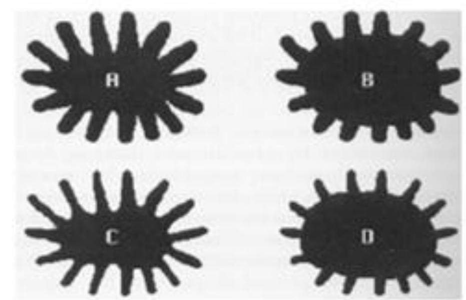
$$\text{Compactness} = \frac{\sqrt{\left(\frac{4}{\pi}\right)\text{Area}}}{\text{Max Diameter}}$$

$$\text{Modification Ratio} = \frac{\text{Inscribed Diameter}}{\text{Maximum Diameter}}$$

$$\text{Extent} = \frac{\text{Net Area}}{\text{Bounding Rectangle}}$$



	Formfactor	Aspect Ratio
A	0.257	1.339
B	0.256	2.005
C	0.459	1.294
D	0.457	2.017



	Roundness	Convexity	Solidity	Compactness
A	0.587	0.351	0.731	0.766
B	0.584	0.483	0.782	0.764
C	0.447	0.349	0.592	0.668
D	0.589	0.497	0.714	0.768

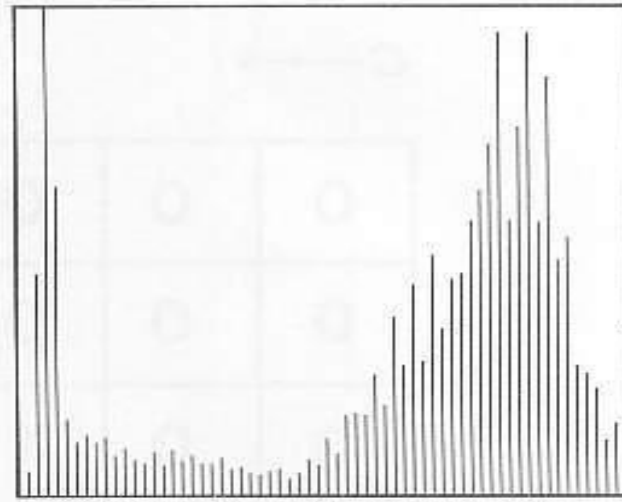
Histogram Features

- The histogram of an image is a plot of the gray-level values versus the number of pixels at that value.

- The shape of the histogram provides us with information about the nature of the image.
  - The characteristics of the histogram has close relationship with characteristic of image such as brightness and contrast.

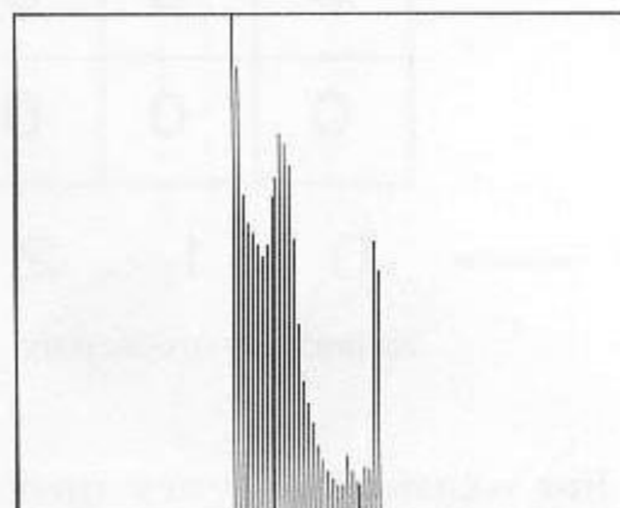
Histogram Features

a. Object in contrast with background.



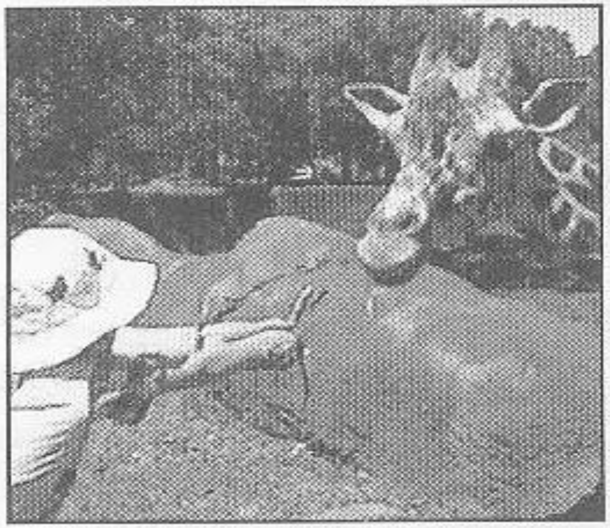
b. Histogram of (a) shows bimodal shape.

c. Low-contrast image.

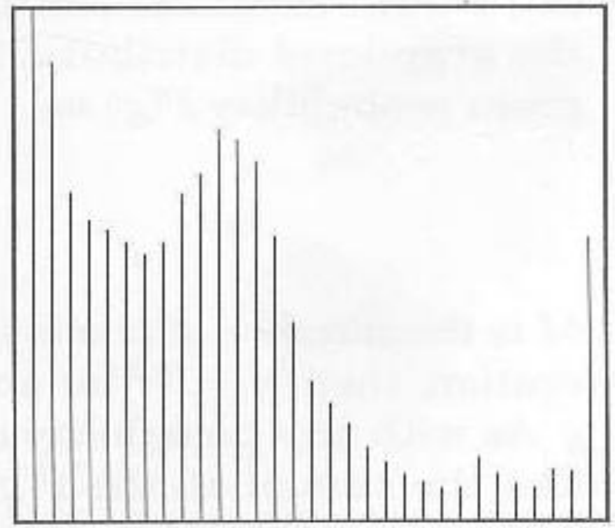


d. Histogram of (c) appears clustered.

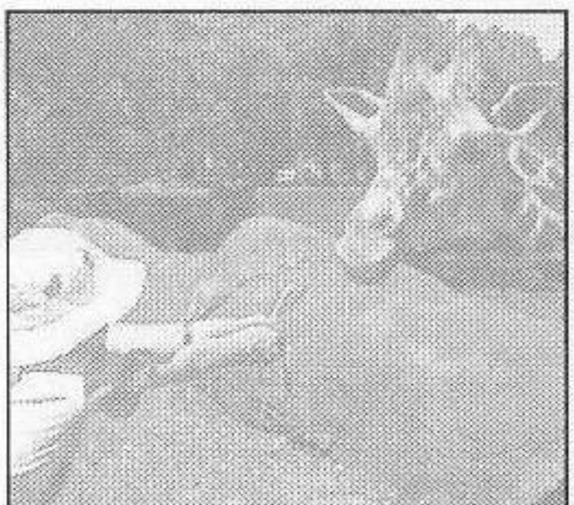
Histogram Features



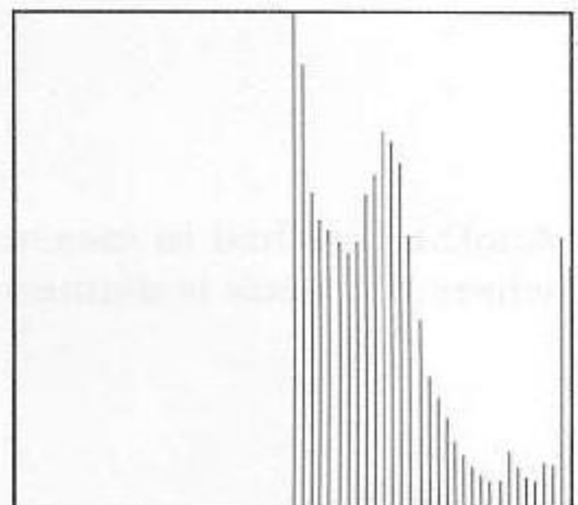
e. High-contrast image.



f. Histogram of (e) appears spread out.



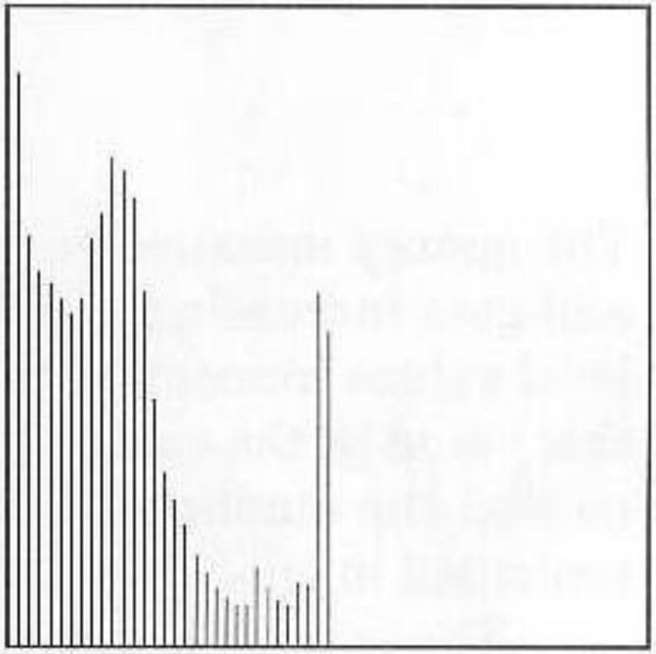
g. Bright image.



h. Histogram of (g) appears shifted to the right.

Histogram Features

i. Dark image.



j. Histogram of (i) appears shifted to the left.

Histogram Features

- The histogram is used as a model of the probability distribution of gray levels.
- The first-order histogram probability *P(g)* is defined as follows:

$$P(g) = \frac{N(g)}{M}$$

P(g) : probability of gray level *g* in image
N(g) : number of pixel with gray level *g* in image
M : total number of pixel in image

Histogram Features

- The features based on the first-order histogram probability are
  - Mean
  - Standard deviation
  - Skew
  - Energy
  - Entropy

Histogram Features – Mean

- The mean is the average value, so it tells us something about the general brightness of the image.
  - A bright image has a high mean.
  - A dark image has a low mean.

- The mean can be defined as follows:

$$Mean = \bar{g} = \sum_{g=0}^{L-1} gP(g) = \sum_{r=0}^{height-1} \sum_{c=0}^{width-1} \frac{I(r,c)}{M}$$

Histogram Features – Standard Deviation

- The standard deviation, which is also known as the square root of the variance, tells something about the contrast.

- It describes the spread in the data.
  - Image with high contrast should have a high standard deviation.

- The standard deviation is defined as follows:

$$\sigma_g = \sqrt{\sum_{g=0}^{L-1} (g - \bar{g})^2 P(g)}$$

Histogram Features – Skew

- The skew measures the asymmetry (unbalance) about the mean in the gray-level distribution.

- Image with bimodal histogram distribution (object in contrast background) should have high standard deviation but low skew distribution (one peak at each side of mean).

Histogram Features – Skew

- Skew can be defined in two ways:

$$SKEW = \frac{1}{\sigma_g{}^3} \sum_{g=0}^{L-1} (g - \bar{g})^3 P(g)$$

$$SKEW' = \frac{\bar{g} - \mathrm{mod}}{\sigma_g}$$

- In the second method, the *mod* is defined as the peak, or highest value.

Histogram Features – Energy

- The energy measure tells us something about how gray levels are distributed.

- The equation for energy is as follows:

$$ENERGY = \sum_{g=0}^{L-1} \left[P(g) \right]^2$$

Histogram Features – Energy

- The energy measure has a value of 1 for an image with a constant value.

- This value gets smaller as the pixel values are distributed across more gray level values.

- A high energy means the number of gray levels in the image is few.
  - Therefore it is easier to compress the image data.

Histogram Features – Entropy

- Entropy measures how many bits do we need to code the image data.
- The equation for entropy is as follows:

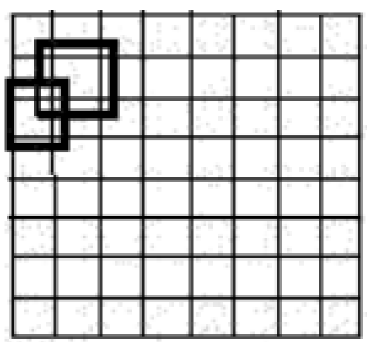
$$ENTROPY = -\sum_{g=0}^{L-1} P(g)\log_2\left[P(g)\right]$$

- As the pixel values are distributed among more gray levels, the entropy increases.

Color Features

Color as low-level feature representation:

- Closely related to human visual perception
  - HSV color model
- Encode the spatial distribution of features in images



- fixed partitioning scheme
- each image divided into M × N overlapping blocks
- 3 separate local histograms (H,S,V) are calculated for every block

- Compact to provide efficient storage and retrieval
  - The location of area-peak for every local histogram determines the value of the corresponding histogram.

Color Features

- Useful in classifying objects based on color.
- Typical color images consist of three color planes: red, green and blue.
  - They can be treated as three separate gray-scale images.
- This approach allows us to use any of the object or histogram features previously defined, but applied to each color band.

Color Features

- However, using absolute color measure such as RGB color space is not robust.
  - There are many factors that contribute to color: lighting, sensors, optical filtering, and any print or photographic process.
  - Any change in these factors will change the absolute color measure.
  - Any system developed based on absolute color measure will not work when any of these factors change.

Color Features

- In practice, some form of relative color measure is best to be used.

- Information regarding relationship between color can be obtained by applying the color transforms defined in Chapter 1.
  - These transforms provide us with two color components and one brightness component.
  - Example: HSL, SCT, Luv, Lab, YCrCb, YIQ, etc.

Spectral Images

- The primary metric for spectral features (frequency-domain-based features) is power.

- Power is defined as the magnitude of the spectral component squared.

$$POWER = \left| T(u,v) \right|^2$$

- Spectral features are useful when classifying images based on textures.
  - Done by looking for peaks in the power spectrum.

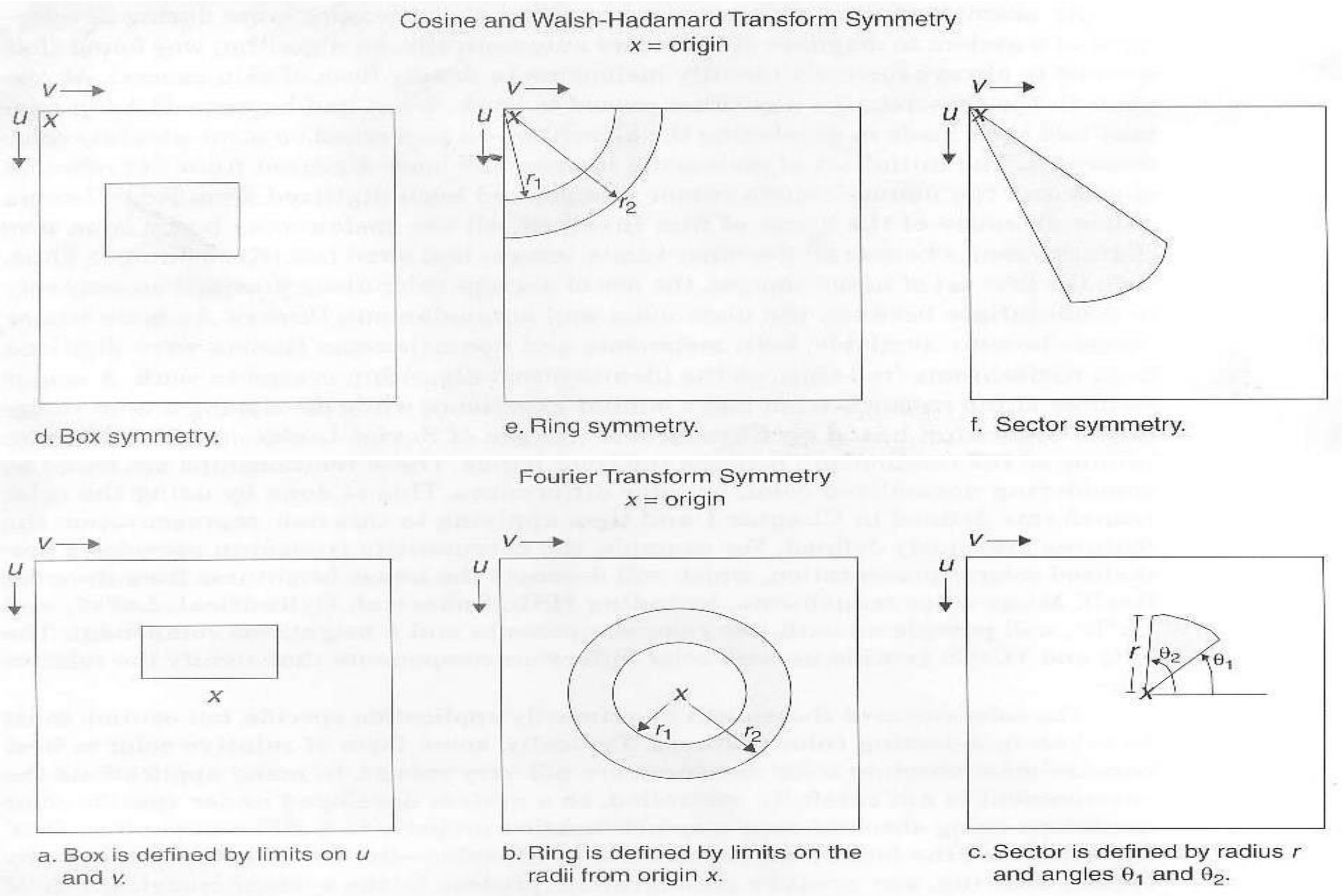
Spectral Images

- It is typical to look at power in various regions, and these regions can be defined as rings, sectors or boxes.

- We can then measure the power in a region of interest by summing the power over the range of frequencies of interest.

$$SpectralRegionPower = \sum_{u \in REGION} \sum_{v \in REGION} \left| T(u,v) \right|^2$$

Spectral Images



Spectral Images

- The ring measure can be used to find texture:
  - High power in small radii corresponds to smooth textures.
  - High power in large radii corresponds to coarse texture.

- The sector power measure can be used to find lines or edges in a given direction, but the results are size invariant.

Feature Analysis

- Important to aid in feature selection process
- Initially, features selected based on understanding of the problem and developer's experience
- FA then will examine carefully to see the most useful & put back through feedback loop
- To define the mathematical tools – feature vectors, feature spaces, distance & similarity measurement

Feature Vectors

- A feature vector is a method to represent an image or part of an image.

- A feature vector is an *n*-dimensional vector that contains a set of values where each value represents a certain feature.

- This vector can be used to classify an object, or provide us with condensed higher-level information regarding the image.

Feature Vector

- Example:

We need to control a robotic gripper that picks parts from an assembly line and puts them into boxes (either box A or box B, depending on object type). In order to do this, we need to determine:

1) Where the object is

2) What type of object it is

The first step would be to define the feature vector that will solve this problem.

Feature Vectors

To determine where the object is:

Use the area and the center area of the object, defined by (r,c).

To determine the type of object:

Use the perimeter of object.

Therefore, the feature vector is: [area, r, c, perimeter]

Feature Vectors

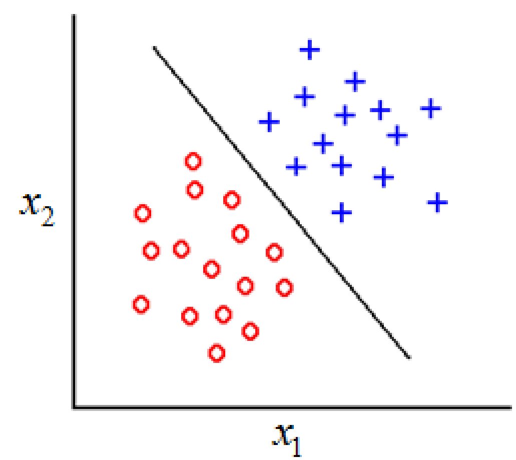
- In feature extraction process, we might need to compare two feature vectors.

- The primary methods to do this are either to measure the difference between the two or to measure the similarity.

- The difference can be measured using a *distance measure* in the n-dimensional space.

Feature Spaces

- A mathematical abstraction which is also *n*-dimensional and is created for a visualization of feature vectors.
- 2-dimensional space:
  - Feature vectors of x_1 and x_2 and two classes represented by *x* and *o*.
  - Each *x* & *o* represents one sample in feature space defined by its values of x_1 and x_2



Distance & Similarity Measures

- Feature vector is to present the object and will be used to classify it
- To perform classification, need to compare two feature vectors
- 2 primary methods – *difference* between two or *similarity*
- Two vectors that are closely related will have small difference and large similarity

~~Distance Measures~~

- Difference can be measured by *distance measure* in *n*-dimensional feature space; the bigger the distance – the greater the difference

- ~~Several metric measurement~~
  - *Euclidean distance*
  - *Range-normalized Euclidean distance*
  - *City block* or *absolute value metric*
  - *Maximum value*

Distance Measures

- *Euclidean distance* is the most common metric for measuring the distance between two vectors.

- Given two vectors **A** and **B**, where:

$$A = \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix}$$
$$B = \begin{bmatrix} b_1 & b_2 & \dots & b_n \end{bmatrix}$$

Distance Measures

- The Euclidean distance is given by:

$$\sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_n - b_n)^2}$$

- This measure may be biased as a result of the varying range on different components of the vector.
  - One component may range 1 to 5, another component may range 1 to 5000.
  - A difference of 5 is significant on the first component, but insignificant on the second component.

Distance Measures

- This problem can be rectified by using *range-normalized Euclidean distance*:

$$\sqrt{\sum_{i=1}^{n} \frac{\left(a_i - b_i\right)^2}{R_i^{\ 2}}}$$

R_i is the range of the *i*th component.

Distance Measures

- Another distance measure, called the *city block* or *absolute value metric*, is defined as follows:

$$\sum_{i=1}^{n} \left| a_i - b_i \right|$$

- This metric is computationally faster than the Euclidean distance but gives similar result.

Distance Measures

- The city block distance can also be range-normalized to give a *range-normalized city block* distance metric, with R_i defined as before:

$$\sum_{i=1}^{n} \left| \frac{a_i - b_i}{R_i} \right|$$

Distance Measures

- The final distance metric considered here is the *maximum value* metric defined by:

$$\max\{|a_1 - b_1|, |a_2 - b_2|, \ldots, |a_n - b_n|\}$$

- The normalized version:

$$\max\{\left|\frac{a_1 - b_1}{R_1}\right|, \left|\frac{a_2 - b_2}{R_2}\right|, \ldots, \left|\frac{a_n - b_n}{R_n}\right|\}$$

Similarity Measures

- The second type of metric used for comparing two feature vectors is the *similarity measure*.

- The most common form of the similarity measure is the vector inner product.

- Using our definition of vector **A** and **B**, the vector inner product can be defined by the following equation:

Similarity Measures

$$\sum_{i=1}^{n} a_i b_i = \left(a_1 b_1 + a_2 b_2 + \ldots + a_n b_n\right)$$

- This similarity measure can also be ranged normalized:

$$\sum_{i=1}^{n} \frac{a_i b_i}{{R_i}^2} = \left(\frac{a_1 b_1}{{R_1}^2} + \frac{a_2 b_2}{{R_2}^2} + \ldots + \frac{a_n b_n}{{R_n}^2}\right)$$

Similarity Measures

- Alternately, we can normalize this measure by dividing each vector component by the magnitude of the vector.

$$\sum_{i=1}^{n} \frac{a_i b_i}{\sqrt{\sum_{j=1}^{n} a_j{}^2} \sqrt{\sum_{j=1}^{n} b_j{}^2}} = \frac{a_1 b_1 + a_2 b_2 + .. + a_n b_n}{\sqrt{\sum_{j=1}^{n} a_j{}^2} \sqrt{\sum_{j=1}^{n} b_j{}^2}}$$

Similarity Measures

- When selecting a feature for use in a computer imaging application, an important factor is the robustness of the feature.

- A feature is robust if it will provide consistent results across the entire application domain.

- For example, if we develop a system to work under any lightning conditions, we do not want to use features that are lightning dependent.

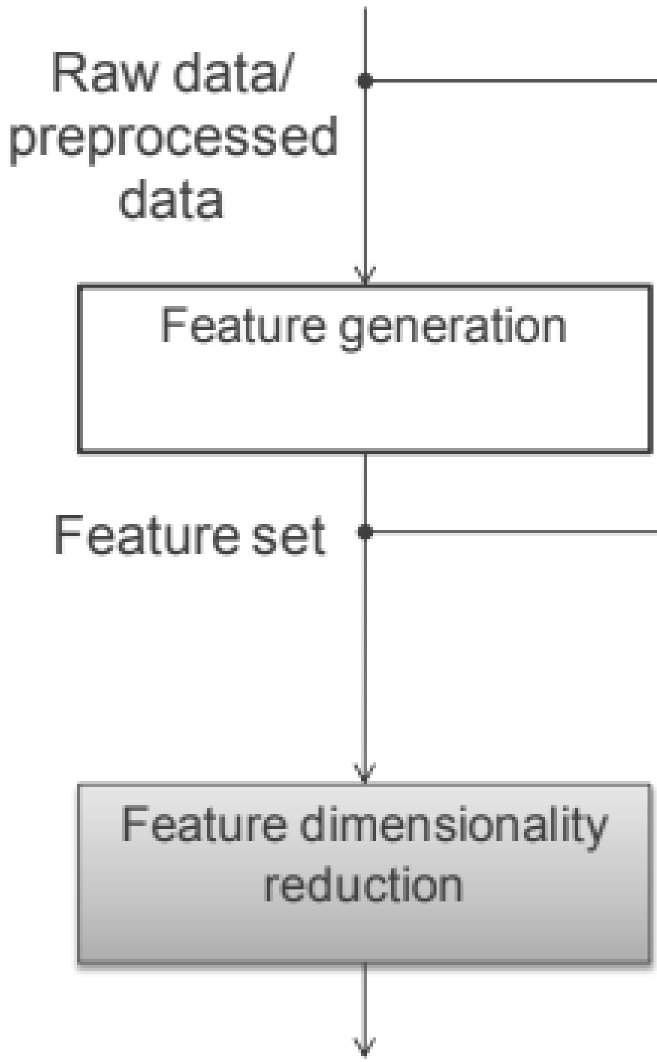
Similarity Measures

- Another type of robustness is called RST-invariance.
  - RST means rotation, size and translation.

- A very robust feature will be RST-invariant.
  - If the image is rotated, shrunk, enlarged or translated, the value of the feature will not change.

Feature extraction/Selection

Feature extraction process



Data representation:
- Varying format: 1-D, 2-D time series, events, ...
- With exhaustive or ad hoc approach
- Dimensionality may reduce or increase
- Incorporate domain knowledge
- Underlying physical phenomenon

Feature representation:
- A scalar or vector per feature
- One vector concatenating all features
- One matrix holding all samples of features

Two approaches
- Select a subset of generated features
- Transform the features to another space with lower dimensions

Features Selection

Feature selection: what are good features

Desired characteristics of features

- High relevance to the objective, e.g., anomaly detection, diagnosis, degradation, PoD/FDR, etc.
- Low redundancy (linearly independent) among the features

Additional characteristic that are frequently overlooked

- Low relevance to non-objective factors, e.g. across assets, environment, usage pattern/ operating conditions, etc.

Features Selection

Feature selection strategies

Filter approach

- Metrics defined using local criteria different from the target models
- Search for 'Good' representation of raw data/features
- Computationally less-expensive

Wrapper

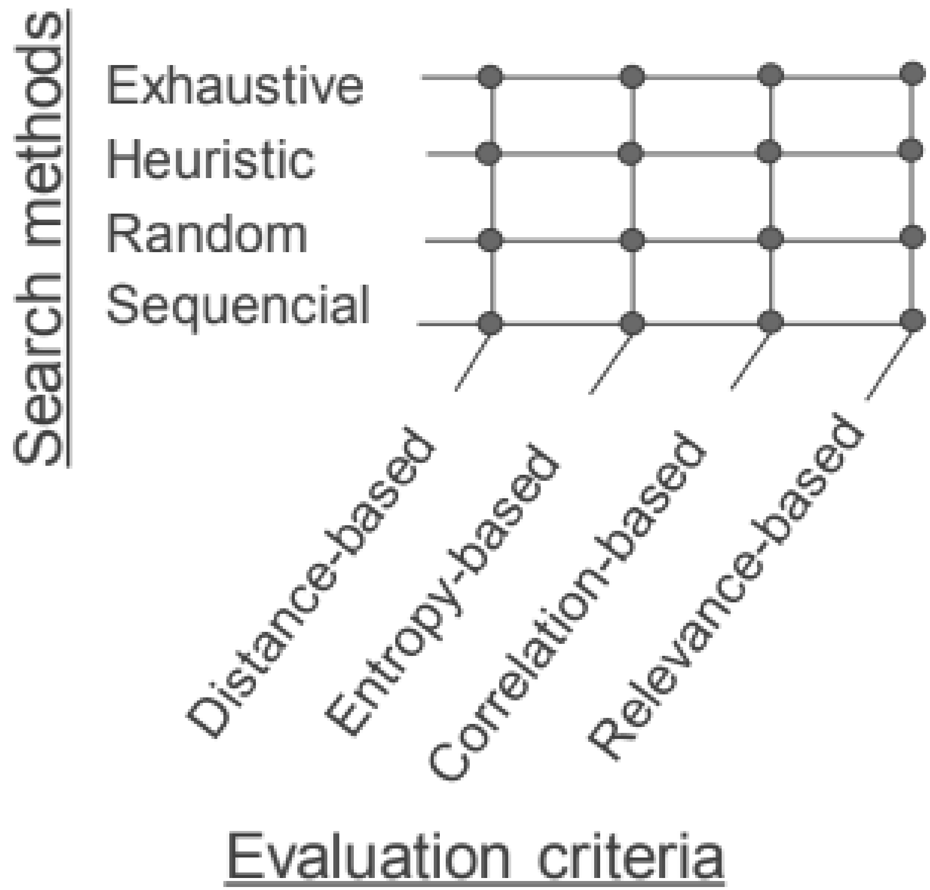
- Metrics defined by the performance (accuracy) of the target models
- 'Application' specific
- Computationally expensive

Embedded approach

- Feature selection built into the target model
- Regression: sparse regression, LASSO, etc.
- Classification: decision tree, regularized random forest

Features Selection

Filter approaches



Examples
- mRMR (Minimum-redundancy-maximum-relevance)
- Fisher score
- Gini score
- Kruskal Wallis statistics

Features Selection

Feature transformation

Linear

- PCA (Principal Component Analysis)
- ICA (Independent component analysis)
- LDA (*Latent Dirichlet Allocation*)
- Latent semantic indexing
- Genetic Programming

Non-linear

- NPCA or KPCA
- NLDA or KLDA
- MDS (Multidimensional scaling)
- Principal curves
- Neural networks
- Genetic Programming

Conclusion

- **Feature Extraction**
  - Binary Object Features (Area, Center of Area, Axis of Least Second Moment, Perimeter, Thinness Ratio, Irregularity, Aspect Ratio, Euler Number, Projection)
  - Histogram Features (Mean, Standard Deviation, Skew, Energy, Entropy)
  - Color Features
  - Spectral Features
- **Feature Analysis**
  - Feature Vectors and Feature Spaces
  – Distance and Similarity Measures (*Euclidean distance, Range-normalized Euclidean distance, City block* or *absolute value metric, Maximum value)*

Conclusion

IMAGE ANALYSIS AND PATTERN RECOGNITION

Feature extraction:

- spatial feature extraction

- transform feature extraction

- edge feature extraction; edge detection

Objects representation by their boundaries:

- contour extraction

- contour descriptors

Objects representation by their regions:

- region extraction

- region representation

Shapes and structures for region-based object representation:

- object skeletons

- binary morphology

- shape descriptors (numerical shape descriptors)

Textures; texture analysis

Image segmentation

Grey level based segmentation/color based segmentation

Connected components analysis

Contour-based segmentation

Region-based segmentation

Mixed techniques

Conclusion

Issues:

- Features have high inconsistent (seemingly noisy) due to
  - Varying operating conditions
  - Asset-to-asset variations
- Features have low sensitivity to faults or degradation

Handling methods

- Normalization / Standardization
- Feature of features (find generalizable features)
- Operating condition clustering & time series segmentation
- Use of local models for post-feature-extraction processing